

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-110678
 (43)Date of publication of application : 22.04.1994

(51)Int.Cl. G06F 9/06
 G06F 9/40
 G06F 9/45

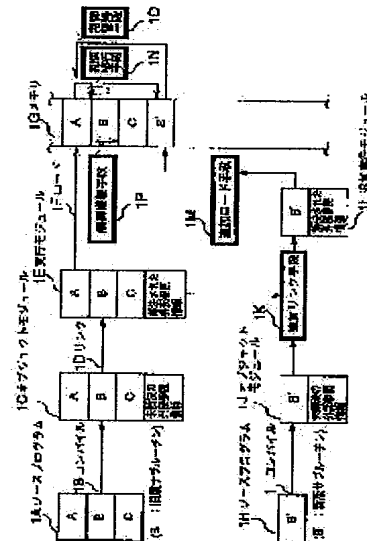
(21)Application number : 04-261796 (71)Applicant : TOSHIBA CORP
 TOSHIBA C EE II SYST KK
 (22)Date of filing : 30.09.1992 (72)Inventor : HIRAYAMA HIDEAKI
 OMORI TAKASHI
 IIJIMA TORU
 KAMEI NOBUYOSHI

(54) PROGRAM EXECUTION SYSTEM AND DYNAMIC CHANGE METHOD FOR EXECUTION PROGRAM

(57)Abstract:

PURPOSE: To replace execution modules without stopping the execution module before applying old bug correction information by dynamically loading a new execution module which is prepared by using the bug correction information on a memory space.

CONSTITUTION: The system features an additional link means (1K) performing the external reference of an object module of a new sub-routine by means of the address information of the execution module and preparing an additional execution module, additional load means (1M) dynamically loading the additional execution module on the memory space during the execution module containing the old sub routines is loaded on the memory and executed, control transition means (1N) passing the control to the new sub routine when the execution module calls the old sub routine, and control recovery means (1O) returning the control to a caller side when the new sub routine ends the processing.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-110678

(43)公開日 平成6年(1994)4月22日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F	9/06	4 4 0 F	9367-5B	
	9/40	3 1 0 B	9189-5B	
	9/45		9292-5B	
			G 0 6 F 9/ 44	3 2 2 K

審査請求 未請求 請求項の数5(全13頁)

(21)出願番号	特願平4-261796	(71)出願人	000003078 株式会社東芝 神奈川県川崎市幸区堀川町72番地
(22)出願日	平成4年(1992)9月30日	(71)出願人	592206145 東芝シーエーイーシステムズ株式会社 神奈川県川崎市幸区堀川町66番2
		(72)発明者	平山 秀昭 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝総合研究所内
		(72)発明者	大森 誉史 神奈川県川崎市幸区小向東芝町1番地 株式会社東芝総合研究所内
		(74)代理人	弁理士 鈴江 武彦

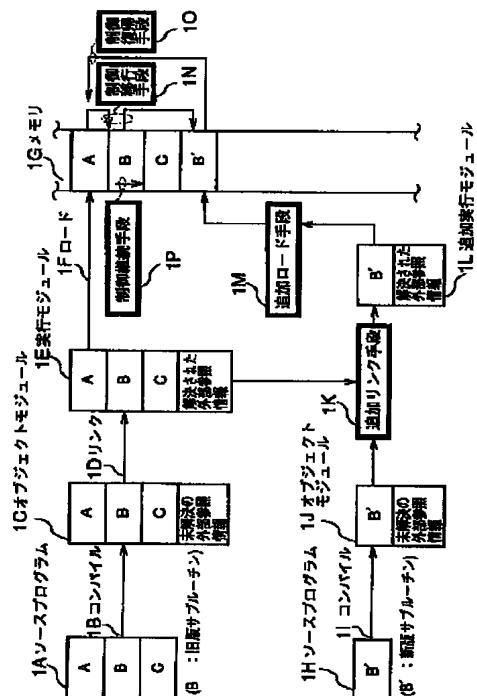
最終頁に続く

(54)【発明の名称】 プログラム実行方式及び実行プログラムの動的変更方法

(57)【要約】

【目的】本発明は、バグ修正情報を用いて作成した新しい実行モジュールをメモリ空間上に動的に追加ロードして、古いバグ修正情報適用前の実行モジュールの停止を待たずに、実行モジュールを入れ換え可能としたことを特徴とする。

【構成】新版のサブルーチンのオブジェクトモジュールの外部参照を、実行モジュールのアドレス情報を用いて解決し、追加実行モジュールを作成する追加リンク手段(1K)と、旧版のサブルーチンを含む実行モジュールがメモリ上にロードされ実行されている際に、上記追加実行モジュールをメモリ空間上に動的に追加ロードする追加ロード手段(1M)と、実行モジュールが旧版のサブルーチン呼び出した場合に、制御を新版のサブルーチンに渡す制御移行手段(1N)と、新版のサブルーチンが処理を終了したとき呼び出し側に制御を戻す制御復帰手段(1O)とを具備してなることを特徴とする。



【特許請求の範囲】

【請求項 1】 ソースプログラムをコンパイルしリンクして実行モジュールを作成し、この実行モジュールをメモリ上にロードして実行する電子計算機システムに於いて、

上記ソースプログラム中の一部のサブルーチンのみを更新しコンパイルして新版のサブルーチンのオブジェクトモジュールを作成し、このオブジェクトモジュールの外部参照を実行モジュールのアドレス情報を用いて決定し、追加実行モジュールを作成する追加リンク手段と、旧版のサブルーチンを含む実行モジュールがメモリ上にロードされ実行されている際に、上記追加実行モジュールをメモリ空間上に動的に追加ロードする追加ロード手段と、

上記実行モジュールが旧版のサブルーチンを呼び出したとき、制御を新版のサブルーチンに渡す制御移行手段と、

上記新版のサブルーチンが処理を終了したとき、制御を呼び出し側に戻す制御復帰手段とを具備してなることを特徴とするプログラム実行方式。

【請求項 2】 実行モジュールが旧版のサブルーチンを呼び出した場合には、制御を新版のサブルーチンに渡す制御移行手段を解消し、再び制御を旧版のサブルーチンに戻す制御移行解消手段を具備したことを特徴とする請求項 1 記載のプログラム実行方式。

【請求項 3】 メモリ上に追加ロードされた、追加実行モジュールからなる新版サブルーチンと、この実行モジュールによって制御を移行された旧版サブルーチンとを世代管理する世代管理手段を具備したことを特徴とする請求項 1 又は 2 記載のプログラム実行方式。

【請求項 4】 指定した時刻から任意の版のサブルーチンへの制御移行手段を適用するための時刻管理手段を具備したことを特徴とする請求項 1 又は 2 又は 3 記載のプログラム実行方式。

【請求項 5】 処理の実行に供されるプログラム中のバグをもつ一部のサブルーチンを対象にバグ修正情報を参照して新版のサブルーチンのオブジェクトモジュールを作成する工程と、

上記オブジェクトモジュールの外部参照を実行モジュールのアドレスを用いて決定し追加実行モジュールを作成する工程と、

旧版のサブルーチンを含む実行モジュールがメモリ上にロードされ実行されている際に上記追加実行モジュールをメモリ空間上に追加ロードする工程と、

上記実行モジュールが旧版のサブルーチンを呼び出した際に制御を上記新版のサブルーチンに受け渡す工程とをもつ実行プログラムの動的変更方法。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 本発明はソフトウェア面で電子計

算機の信頼性を向上させるための電子計算機のプログラム実行方式に係り、特に、バグ修正情報適用前の実行モジュールの停止を待たずに修正された追加実行モジュールの入れ替えを可能にしたプログラム実行方式及び実行プログラムの動的変更方法に関する。

【0002】

【従来の技術】 現在の OS、DBMS、トランザクション処理システム等の大規模ソフトウェアは、様々な処理要求に対応するため、あるいは高度なプロセッサ、デバイスに対応するため、数百万行の規模に達している。

【0003】 しかしソフトウェア開発技術は、数百万行の規模のプログラムから、バグを取り除くのに十分な手法を持ち合わせてはいない。そのため、運用中のプログラムでさえ、多数のバグが内在されたまま、実行を継続している。

【0004】 通常、このようなバグは、適時に発見され、バグ修正情報が作成され、プログラムに適用され、コンパイル/リンクされ、新しい実行モジュールが作成される。新しい実行モジュールは、古いバグ修正情報適用前の実行モジュールの停止を待って、入れ換えが行なわれる。

【0005】 しかし、OS、DBMS、トランザクション処理システムといった長時間実行されるシステムでは、実行モジュールの停止の機会が少なく、従って実行モジュールの入れ換えの機会が少なくなり、バグ修正情報が作成されたとしても、実際にそのバグ修正情報が適用し難くなっている。

【0006】 今後はシステムの無停止稼働の機会が増え、上記したような問題は益々深刻となることが予想される。フォールトトレラントシステム (fault-tolerant system) によって、ハードウェア故障に起因したシステムダウンは減少させることができたが、上記したようなソフトウェアバグに起因したシステムダウンを減少させることはできず、しかも処理の高度化と、システムの無停止稼働に伴って、ソフトウェアバグの問題は益々深刻になってきている。

【0007】

【発明が解決しようとする課題】 このように従来のソフトウェア開発環境に於いては、プログラムのバグを発見し、バグ修正情報を作成し、コンパイル/リンクして新しい実行モジュールを作成したとしても、古いバグ修正情報適用前の実行モジュールの停止を待って、実行モジュールを入れ換えなければならず、バグ修正情報の適用には、システムの停止を伴わなければならないという問題があった。

【0008】 本発明は上記したような従来技術の欠点を除去し、プログラムのバグを発見し、バグ修正情報を作成し、コンパイル/リンクして新しい実行モジュールを作成した場合に、古いバグ修正情報適用前の実行モジュールの停止を待たずに、実行モジュールを入れ換え、バ

グ修正情報の適用に、システムの停止を伴うことを不要とした、プログラム実行方式を提供することを目的とする。

【0009】

【課題を解決するための手段】上記目的を達成するために、本発明に於いては、ソースプログラムをコンパイル／リンクして実行モジュールを作成し、この実行モジュールをメモリ上にロードして実行する電子計算機システムに於いて、

【0010】上記プログラム中の一部のサブルーチンのみを更新／コンパイルして新版のサブルーチンのオブジェクトモジュールを作成し、このオブジェクトモジュールの外部参照を、実行モジュールのアドレス情報を用いて解決し、追加実行モジュールを作成する追加リンク手段（図1符号1k、及び図2参照）と、

【0011】旧版のサブルーチンを含む実行モジュールがメモリ上にロードされ実行されている中に、上記追加実行モジュールをメモリ空間上に動的に追加ロードする追加ロード手段（図1符号1M、及び図3参照）と、実行モジュールが旧版のサブルーチンを読み出した場合に、制御を新版のサブルーチンに渡す制御移行手段（図1符号1N、及び図4参照）と、

【0012】上記新版のサブルーチンが処理を終了して呼び出し側に制御を戻す際に正常に制御を呼び出し側に戻す制御復帰手段（図1符号1O参照）と、旧版のサブルーチン中で実行中の処理はそのまま処理を継続する制御継続手段（図1符号1P参照）とを具備してなることを特徴とする。

【0013】更に本発明は、実行モジュールが旧版のサブルーチンを読み出した場合に、制御を新版のサブルーチンに渡す制御移行手段を解消し、再び制御を旧版のサブルーチンに戻す制御移行解消手段（図5符号5Q参照）を付加したことを特徴とする。

【0014】更に本発明は、メモリ上に追加ロードされた追加実行モジュールからなる新版サブルーチンと、実行モジュールによって制御を移行された旧版サブルーチンを世代管理する世代管理手段（図6符号6R参照）を付加したことを特徴とする。

【0015】更に本発明は、指定した時刻から、任意の版のサブルーチンへの制御移行手段を適用するための時刻管理手段（図7符号7S参照）を付加したことを特徴とする。

【0016】

【作用】上記構成に於いて、追加リンク手段（1k）は、バグ修正情報をもとに作成された新版のサブルーチンのオブジェクトモジュール（図1符号1j）の外部参照を実行モジュールのアドレス情報を用いて決定（解決）し、追加実行モジュール（1l）を作成する。この追加実行モジュールは追加ロード手段（1m）により、旧版のサブルーチンを含む実行モジュールがメモリ上にロー

ドされ実行されている際に、メモリ空間上に動的に追加ロードされる。そして実行モジュールが旧版のサブルーチンを読み出した際に、制御移行手段（1n）により、制御が新版のサブルーチンに渡される。又、上記新版のサブルーチンが処理を終了する際は、制御復帰手段（1o）により、制御が呼び出し側に戻される。又、旧版のサブルーチン中で実行中の処理は制御継続手段（1p）により、そのまま処理が継続される。

【0017】このようにして、プログラムのバグを発見し、バグ修正情報を作成し、コンパイル／リンクして新しい実行モジュールを作成した場合に、古いバグ修正情報適用前の実行モジュールの停止を待たずに実行モジュールを入れ換えるプログラム実行手段が実現される。これにより、バグ修正情報の適用に、システムの停止を伴うことを不要としたプログラム実行方式が実現できる。

【0018】又、本発明によれば、上記バグ修正情報が不適切であったとき、古いバグ修正情報適用前の実行モジュールに制御を戻す際に、システムの停止を伴うことを不要としたプログラム実行方式が実現できる。

【0019】

【実施例】以下図面を参照して本発明の一実施例を説明する。図1は本発明の第1実施例の構成を示す図である。

【0020】ここでは、3つのサブルーチン[A]、[B]、[C]からなるソースプログラム（1A）をコンパイル（1B）し、オブジェクトモジュール（1C）を作成し、更にそれをリンク（1D）して、実行モジュール（1E）を作成し、それをメモリ空間（1G）上にロード（1F）し、実行している。

【0021】ここで、ソースプログラム（1A）中のサブルーチン[B]にバグが発見されると、そのバグ修正情報を作成し、更にバグ修正情報を用いてサブルーチン[B]を修正し、新たなサブルーチン[B']を作成する。

【0022】このサブルーチン[B']が作成されると、サブルーチン[B']を含むソースプログラム（1H）をコンパイル（1I）して、オブジェクトモジュール（1J）を作成する。

【0023】このオブジェクトモジュール（1J）中のサブルーチン[B']からの外部参照が、オブジェクトモジュール（1C）中のサブルーチン[B]からの外部参照と同様になるように、追加リンク手段（1K）によって、追加実行モジュール（1L）を作成する。

【0024】この追加実行モジュール（1L）は、追加ロード手段（1M）によって、プログラム[A]、[B]、[C]を含む実行モジュール（1E）がロードされているメモリ空間（1G）上に、[B']を他と重ならないようロードする。

【0025】更に、メモリ空間（1G）上に既にロードされている、実行モジュール[B]のエントリーポイント

のマシン命令を、トラップ命令、あるいは不正命令に置き換える。

【0026】これにより、以降に実行モジュール[B]が呼び出された場合は、[B]のエントリポイントのマシン命令が、トラップ命令、又は不正命令に置き換えられているので、トラップが発生する。

【0027】トラップが発生した場合は、制御移行手段(1N)により、実行モジュール[B]の代わりに、サブルーチン[B']が呼ばれたかのように、制御を[B']に移す。サブルーチン[B']の実行が終了した場合には、制御復帰手段(1O)によって、呼び出した側に制御を戻す。

【0028】尚、追加実行モジュール(1L)をメモリ空間(1G)上に追加ロードしたとき、サブルーチン[B]が実行中であった場合には、サブルーチン[B]の実行は継続しなければならない。そのために、メモリ空間(1G)上にロードされていたサブルーチン[B]は、エントリポイントを除き、制御継続手段(1P)によって、そのままの状態にしておく。

【0029】上記した手段により、追加実行モジュール(1L)をメモリ空間(1G)上に追加ロードしたとき、サブルーチン[B]が実行中であったとしても、サブルーチン[B]が呼び出し側に制御を戻すまでは、サブルーチン[B]が実行され続けるが、サブルーチン[B]が呼び出し側に制御を戻した以降は、サブルーチン[B']が実行されるようになり、長時間、実行を続けるシステムに於いても、システムを停止させることなく、プログラムのバグ修正を行なうことが可能となる。

【0030】図2は本発明が適用されるプログラム実行方式に於いて、バグ修正情報を適用したサブルーチンのオブジェクトモジュールの外部参照を、基のサブルーチンを含む実行モジュールを用いて解決する、追加リンクの処理の流れを説明するフローチャートである。

【0031】図2に示す追加リンク手段(1K)の処理に於いては、先ずオブジェクトモジュール(1J)中に未解決の外部参照が存在するか否かを検査する(ステップ2A)。ここで、未解決の外部参照情報が存在するとき(ステップ2A-Yes)は、既に存在する実行モジュール(1E)の外部参照情報に合わせて外部参照を解決(決定)する(ステップ2B)。そして、上記ステップ2Aの処理に戻る。全ての外部参照を解決したならば(ステップ2A-No)、解決された外部参照情報に基づいて、追加実行モジュール(1L)を作成する。

【0032】図3は本発明が適用されるプログラム実行方式に於いて、上記追加リンク手段(1K)によって作成された追加実行モジュールを、既に実行モジュールがロードされているメモリ空間上に追加ロードする、追加ロード処理の流れを説明するフローチャートである。

【0033】図3に示す追加ロード手段(1M)の処理に於いては、現在、実行されている実行モジュール(1

E)がロードされているメモリ空間上に、追加実行モジュール(1L)をロードできる大きさの空き領域を探す(ステップ3A)。そして、探した空き領域に、追加実行モジュール(1L)をロードする(ステップ3B)。

【0034】次に、追加実行モジュール(1L)によって機能を置き換えられるメモリ空間上に既にロードされているサブルーチンのエントリポイントのマシン命令をトラップ命令あるいは不正命令に置き換える(ステップ3C)。

【0035】図4は本発明が適用されるプログラム実行方式に於いて、上記追加ロード手段によって、既に実行モジュール(1E)がロードされているメモリ空間上に追加実行モジュール(1L)を追加ロードして処理を再開した後、バグ修正情報を適用する前のサブルーチンが呼び出された場合に、制御をバグ修正情報を適用したサブルーチンに移行する際の処理の流れを説明するフローチャートである。

【0036】図4に示す制御移行手段(1N)の処理に於いては、発生したトラップを検出し(ステップ4A)、検出したトラップが、バグ修正情報を適用したサブルーチンによって置き換えられたサブルーチンのエントリポイントで発生したか否かを検査する(ステップ4B)。

【0037】ここで、検出したトラップがバグ修正情報を適用したサブルーチンによって置き換えられたサブルーチンのエントリポイントで発生したものでないときは通常のトラップ処理を実行する(ステップ4C)。

【0038】又、検出したトラップがバグ修正情報を適用したサブルーチンによって置き換えられたサブルーチンのエントリポイントで発生したものであるときは制御を新しいサブルーチンに移す(ステップ4D)。

【0039】尚、上記した第1実施例に於いては、制御復帰手段、制御継続手段は特に設定せずとも、追加ロード手段(1M)、制御移行手段(1N)がその一部処理機能として実現している。

【0040】本発明の第2実施例として、追加ロード手段(1M)に於いて、追加実行モジュール(1L)によって機能を置き換えられる、メモリ空間上に既にロードされているサブルーチンのエントリポイントのマシン命令を、トラップ命令あるいは不正命令に代わり、無条件分岐命令に置き換える。

【0041】この場合は、制御移行手段(1N)も、より単純になり、バグ修正情報を適用する前のサブルーチンが呼び出された場合に、トラップ操作を介さずに、直接、制御をバグ修正情報を適用したサブルーチンに移行することが可能となる。但しこの手段による実施の可能性は、無条件分岐命令が条件コード(CONDITION CODE)を変更しない等、マシンの特性に依存する。

【0042】本発明の第3実施例として、追加ロード手

段(1M)に於いて、追加実行モジュール(1L)によって機能を置き換えられる、メモリ空間上に既にロードされているサブルーチンのエントリポイントのマシン命令を、トラップ命令あるいは不正命令に代わり、上記サブルーチン呼び出す側の分岐先アドレスを書き換える手段に置き換える。

【0043】この場合は、制御移行手段(1N)も、より単純になり、バグ修正情報を適用する前のサブルーチン呼び出す代わりに、バグ修正情報を適用したサブルーチン呼び出すことになる。但しこの手段による実施の可能性は、サブルーチンのエントリポイントを変数として格納しない等、言語の特性に依存する。

【0044】本発明の第4実施例として、追加ロード手段(1M)に於いて、追加実行モジュール(1L)によって機能を置き換えられる、メモリ空間上に既にロードされているサブルーチンのエントリポイントのマシン命令をトラップ命令あるいは不正命令に代わり、上記サブルーチン呼び出すための分岐先アドレスを格納したテーブルを書き換える手段に置き換える。

【0045】この場合に於いても、制御移行手段(1N)が単純になり、バグ修正情報を適用する前のサブルーチン呼び出す代わりに、バグ修正情報を適用したサブルーチン呼び出すことになる。但しこの手段を実施するためには、全てのサブルーチンのエントリポイントをテーブル内に格納し、間接呼び出しを行なわなければならない、プログラム構築法に依存する。図5は本発明の第5実施例を示す構成図である。

【0046】この第5実施例では、上記した図1に示す第1実施例に、更に、旧版のサブルーチンから新版のサブルーチンに渡した制御を、再び旧版のサブルーチンに戻す、制御移行解消手段(5Q)を加えている。

【0047】新版のサブルーチンに於ける修正が不適切であった場合には、この制御移行解消手段(5Q)を使用することにより、新版のサブルーチンへの移行を中止し、再び旧版のサブルーチンに制御を戻すことが可能となる。図6は本発明の第6実施例を示す構成図である。

【0048】この第6実施例では、上記した第5実施例に、更に、追加実行モジュールからなる新版のサブルーチンと、この実行モジュールによって制御を移行された旧版サブルーチンを世代管理する世代管理手段(6R)を加えている。

【0049】修正を加えたサブルーチンに更に修正を加えたり、新版のサブルーチンに於ける修正が数世代に渡って不適切であった場合には、世代管理手段(6R)を使用することにより、新版のサブルーチンへの移行を中止し、数世代に渡った旧版のサブルーチンに制御を戻すことが可能となる。図7は本発明の第7実施例を示す構成図である。

【0050】この第7実施例では、上記した第6実施例に、更に、指定した時刻から、任意のサブルーチンへの

制御移行手段を適用するための時刻管理手段(7S)を加えている。

【0051】新版のサブルーチンへの移行時刻や、旧版のサブルーチンへの復帰時刻を指定する場合には、時刻管理手段(7S)を使用することにより、指定した時刻から、任意の版のサブルーチンへの制御移行手段を適用することが可能となる。

【0052】

【発明の効果】以上詳記したように本発明によれば、プログラムのバグを発見しバグ修正情報を作成し、コンパイル/リンクして新しい実行モジュールを作成した場合に、古いバグ修正情報適用前の実行モジュールの停止を待たずに、実行モジュールを入れ換え、バグ修正情報の適用に、システムの停止を伴うことを不要としたプログラム実行方式が実現できる。更にこのバグ修正情報が不適切であった場合に、古いバグ修正情報適用前の実行モジュールに制御を戻す際に、システムの停止を伴うことを不要としたプログラム実行方式が実現できる。

【図面の簡単な説明】

【図1】本発明の第1実施例を説明するための全体の構成を示す図。

【図2】図1の実施例に於ける追加リンク処理を説明するための流れ図。

【図3】図1の実施例に於ける追加ロード処理を説明するための流れ図。

【図4】図1の実施例に於ける、制御をバグ修正情報を適用したサブルーチンに移行する処理を説明するための流れ図。

【図5】本発明の第5実施例を説明するための全体の構成を示す図。

【図6】本発明の第6実施例を説明するための全体の構成を示す図。

【図7】本発明の第7実施例を説明するための全体の構成を示す図。

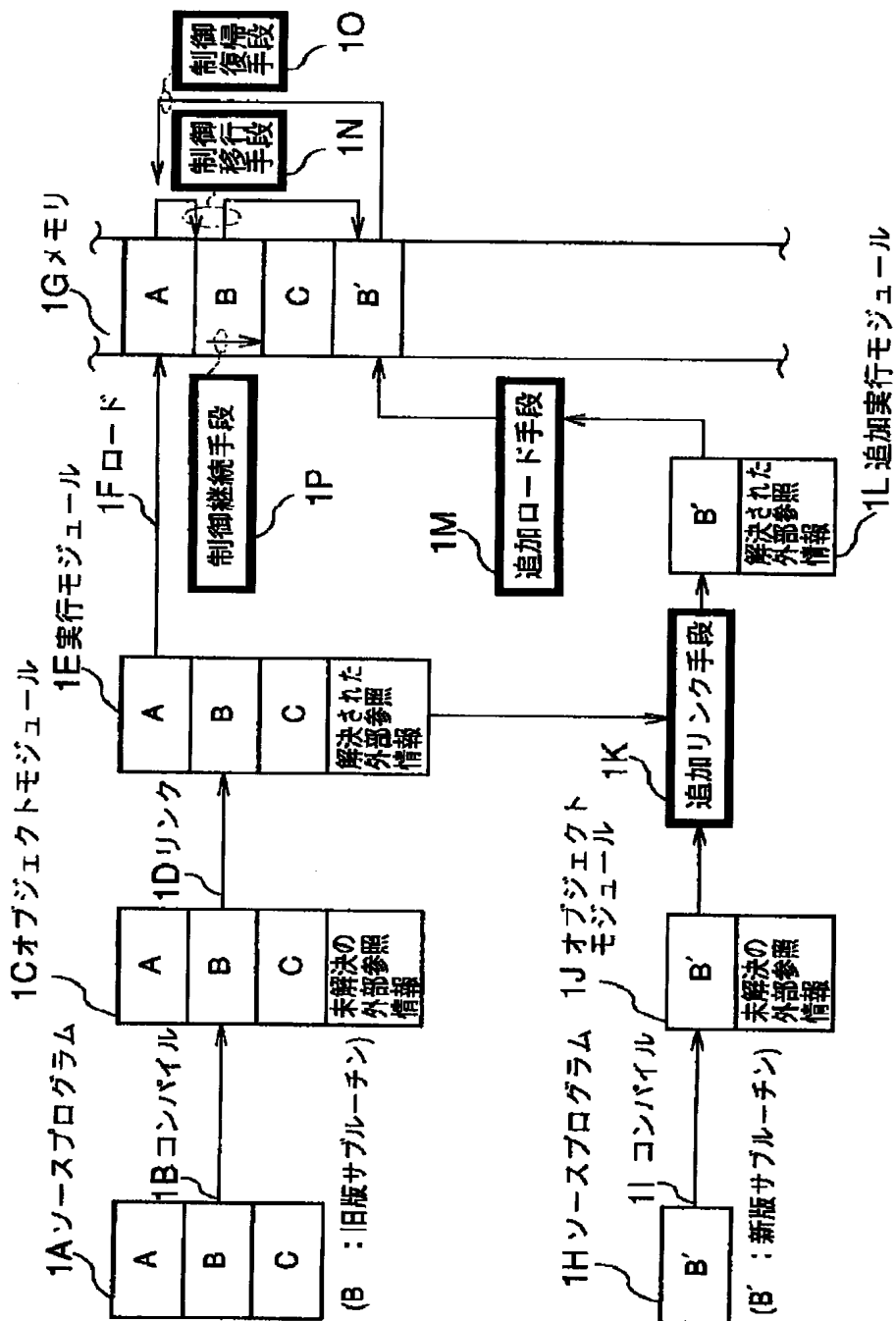
【符号の説明】

1A…サブルーチン[A], [B], [C]を含むソースプログラム、1B…コンパイル手段、1C…サブルーチン[A], [B], [C]を含むオブジェクトプログラム、1D…リンク手段、1E…サブルーチン[A], [B], [C]を含む実行モジュール、1F…実行モジュールをメモリ上にロードするロード手段、1G…実行モジュールのロードされたメモリ空間、1H…サブルーチン[B]にバグ修正情報を適用したサブルーチン[B']を含むソースプログラム、1I…コンパイル手段、1J…サブルーチン[B]にバグ修正情報を適用したサブルーチン[B']を含むオブジェクトモジュール、1K…追加リンク手段、1L…サブルーチンBにバグ修正情報を適用したサブルーチン[B']を含む追加実行モジュール、1M…追加ロード手段、1N…制御移行手段、1O…制御復帰手段、1P…制御継続手段、5

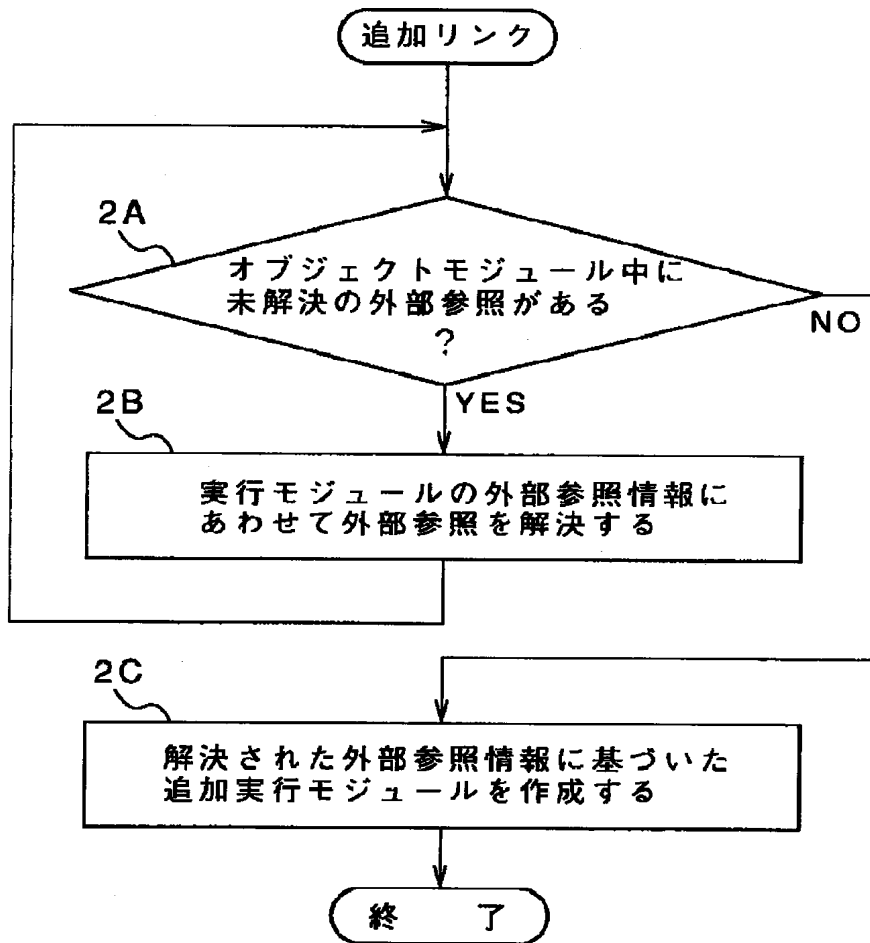
A…サブルーチン [A], [B], [C] を含むソースプログラム、5B…コンパイルするための手段、5C…サブルーチン [A], [B], [C] を含むオブジェクトプログラム、5D…リンク手段、5E…サブルーチン [A], [B], [C] を含む実行モジュール、5F…実行モジュールをメモリ上にロードするロード手段、5G…実行モジュールのロードされたメモリ空間、5H…サブルーチンBにバグ修正情報を適用したサブルーチン [B′] を含むソースプログラム、5I…コンパイル手段、5J…サブルーチン [B] にバグ修正情報を適用したサブルーチン [B′] を含むオブジェクトモジュール、5K…追加リンク手段、5L…サブルーチン [B] にバグ修正情報を適用したサブルーチン [B′] を含む追加実行モジュール、5M…追加ロード手段、5N…制御移行手段、5O…制御復帰手段、5P…制御継続手段、5Q…制御移行解消手段、6A…サブルーチン [A], [B], [C] を含むソースプログラム、6B…コンパイル手段、6C…サブルーチン [A], [B], [C] を含むオブジェクトプログラム、6D…リンク手段、6E…サブルーチン [A], [B], [C] を含む実行モジュール、6F…実行モジュールをメモリ上にロードするロード手段、6G…実行モジュールのロードされたメモリ空間、6H…サブルーチン [B] にバグ修正情報を適用したサブルーチン [B′] を含むソースプログラム、6I…コンパイル手段、6J

…サブルーチン [B] にバグ修正情報を適用したサブルーチン [B′] を含むオブジェクトモジュール、6K…追加リンク手段、6L…サブルーチン [B] にバグ修正情報を適用したサブルーチン [B′] を含む追加実行モジュール、6M…追加ロード手段、6N…制御移行手段、6O…制御復帰手段、6P…制御継続手段、6Q…制御移行解消手段、6R…世代管理手段、7A…サブルーチン [A], [B], [C] を含むソースプログラム、7B…コンパイル手段、7C…サブルーチン [A], [B], [C] を含むオブジェクトプログラム、7D…リンク手段、7E…サブルーチン [A], [B], [C] を含む実行モジュール、7F…実行モジュールをメモリ上にロードするロード手段、7G…実行モジュールのロードされたメモリ空間、7H…サブルーチン [B] にバグ修正情報を適用したサブルーチン [B′] を含むソースプログラム、7I…コンパイル手段、7J…サブルーチン [B] にバグ修正情報を適用したサブルーチン [B′] を含むオブジェクトモジュール、7K…追加リンク手段、7L…サブルーチン [B] にバグ修正情報を適用したサブルーチン [B′] を含む追加実行モジュール、7M…追加ロード手段、7N…制御移行手段、7O…制御復帰手段、7P…制御継続手段、7Q…制御移行解消手段、7R…世代管理手段、7S…時刻管理手段。

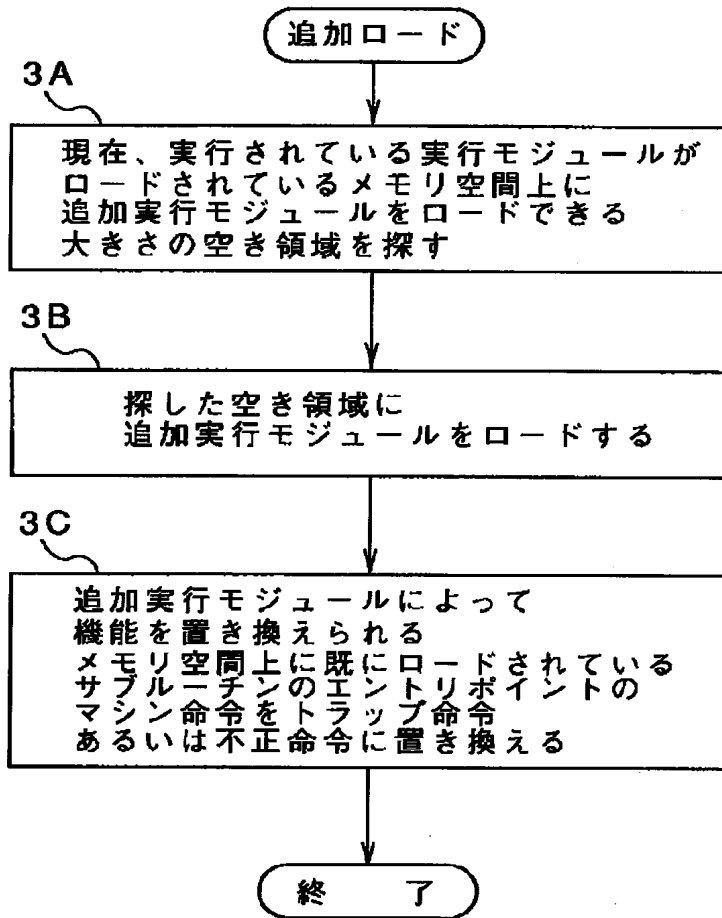
【図1】



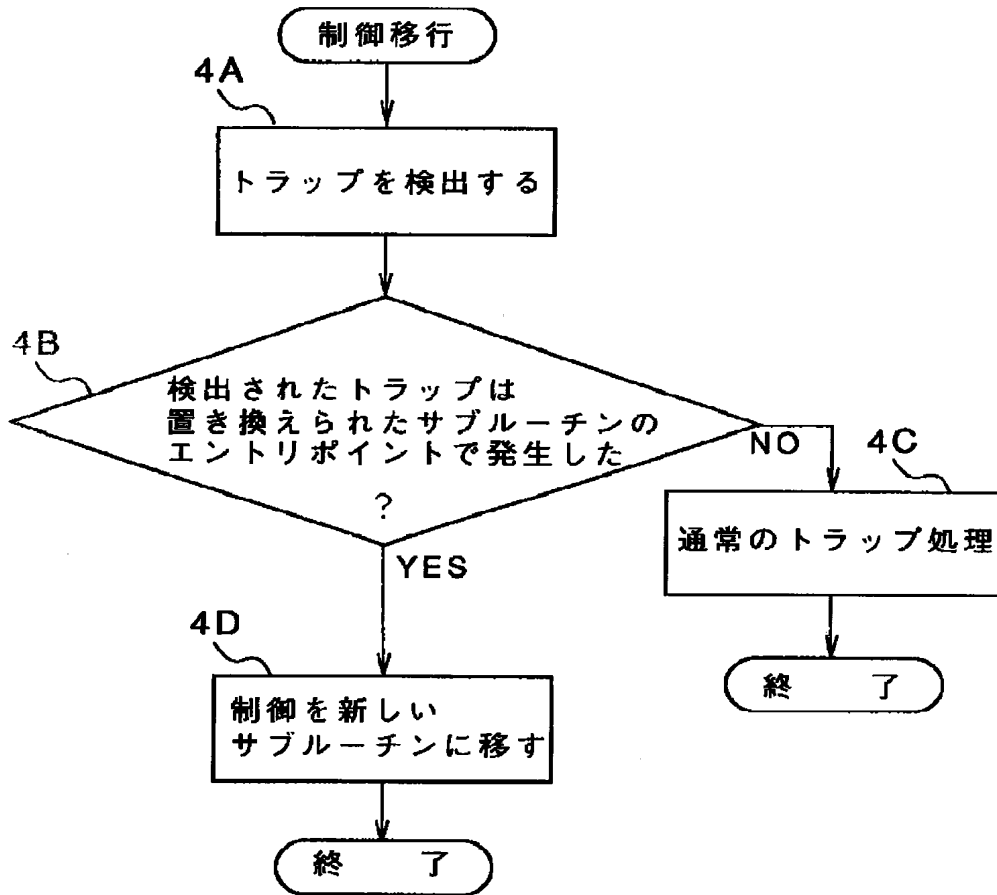
【図2】



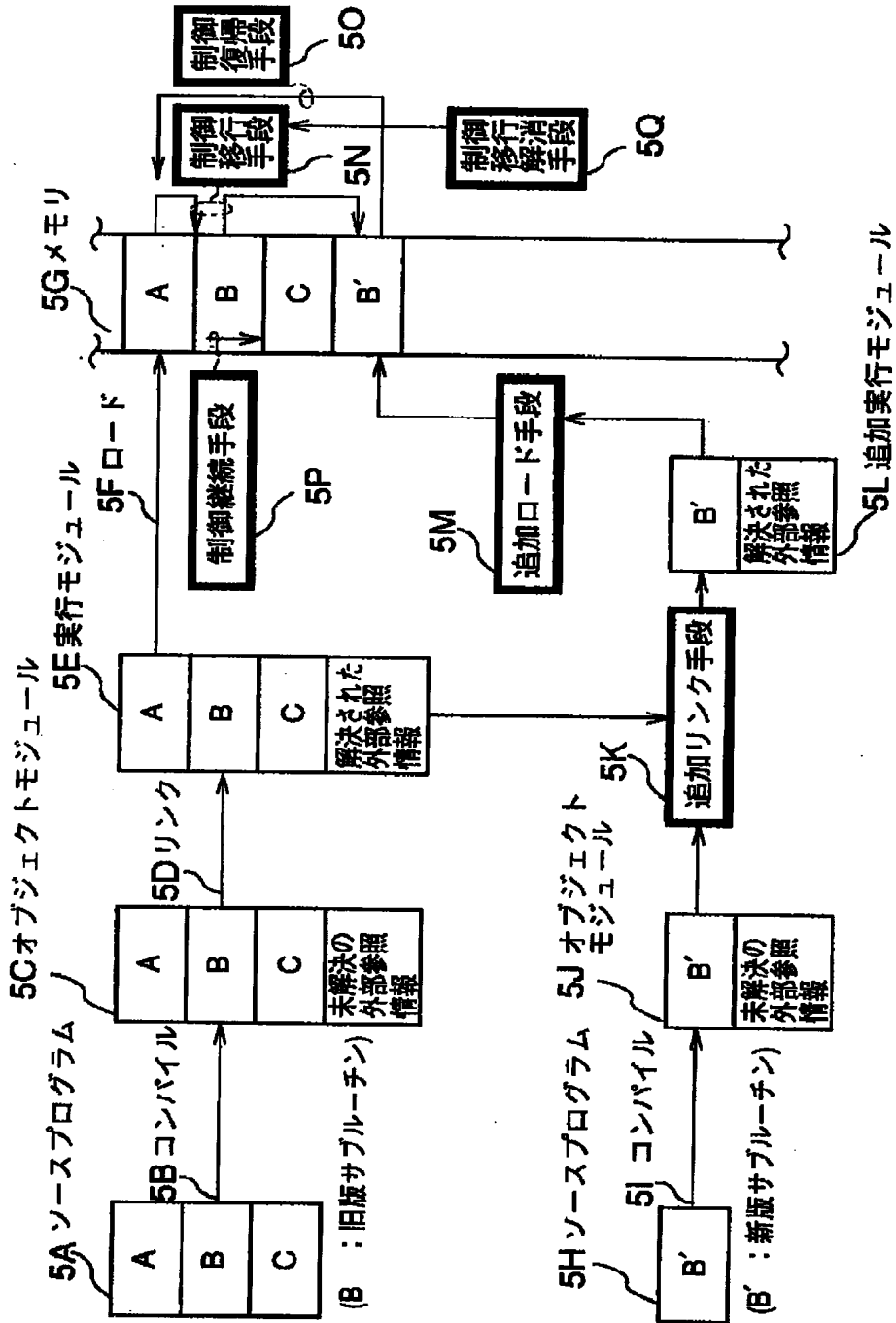
【図3】



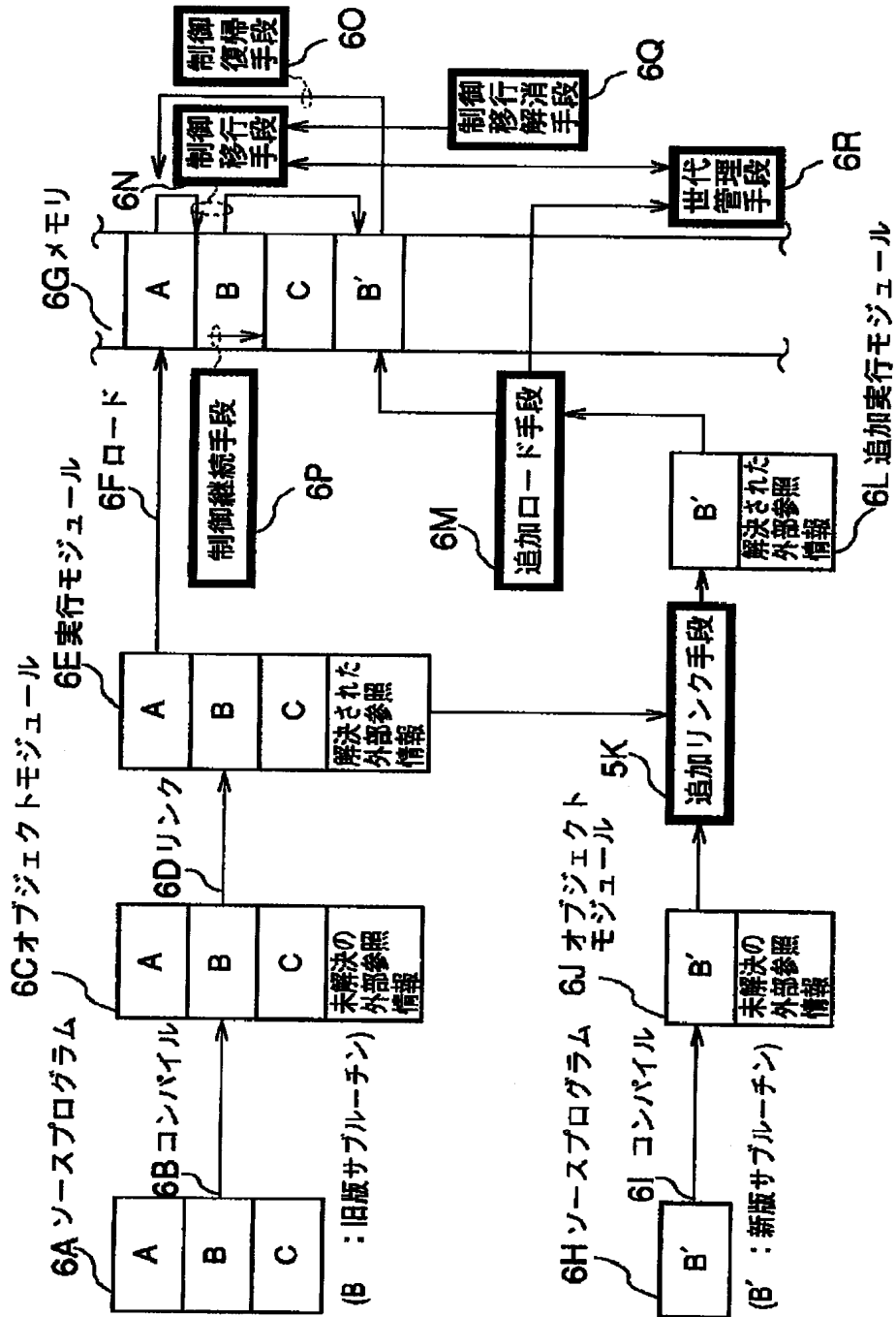
【図4】



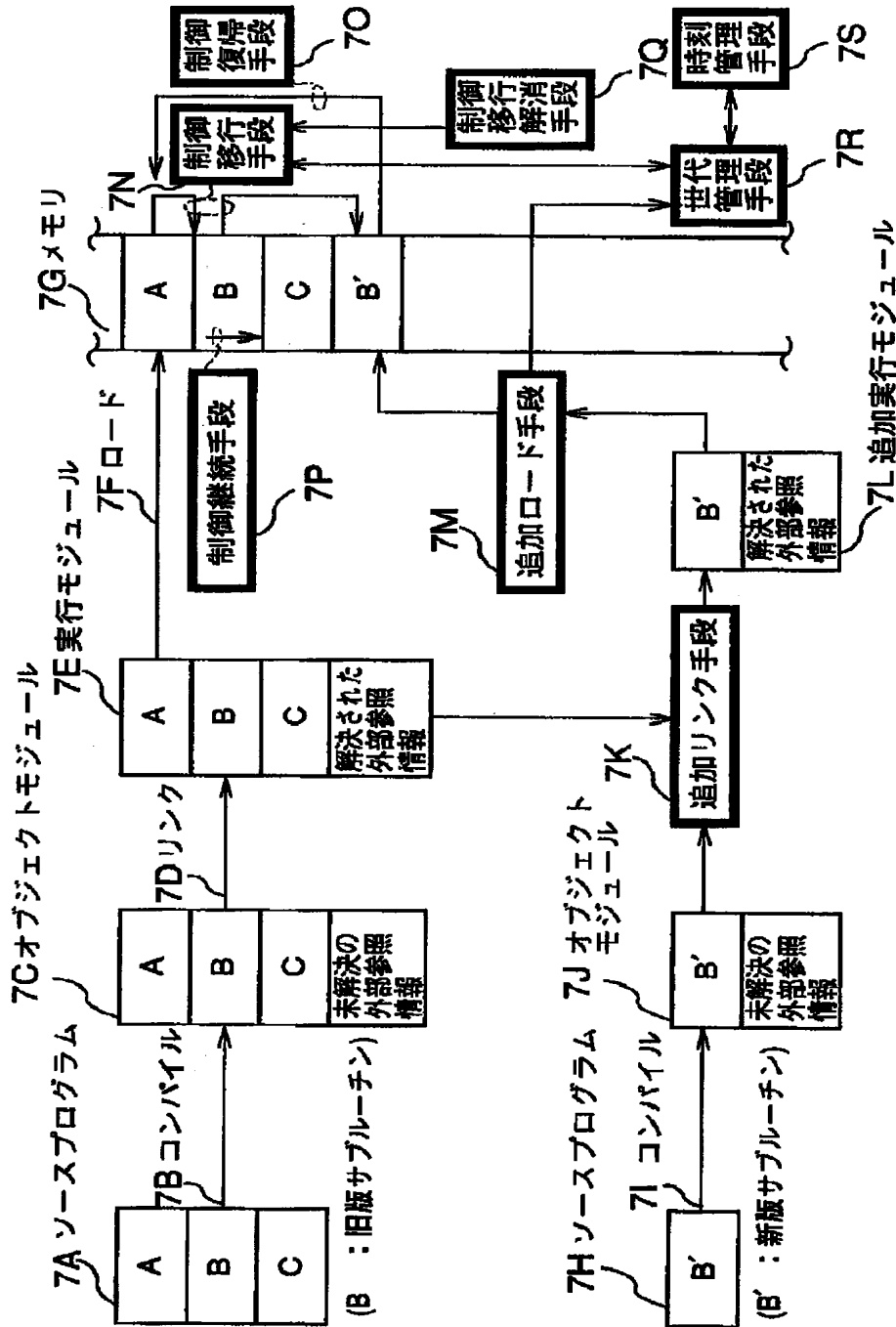
【図5】



【図6】



【図7】



フロントページの続き

(72) 発明者 飯島 徹
 神奈川県川崎市幸区小向東芝町1番地 株
 式会社東芝総合研究所内

(72) 発明者 亀井 信義
 神奈川県川崎市幸区堀川町66番2 東芝シ
 ーエーイーシステムズ株式会社内